**CSE5231 COMPUTER Network**

**Project**

Chia-Che Lee, Sravani Annapareddy, Weichun Xu, Yutian Gui

1. **Description of Problem and Algorithm**

In the design (figure 1), Node A will send packets to Node B and Node B will send ACKs back to Node A. Sliding window is used in both Node A and Node B to constrain the number of the packets sent between these 2 nodes.

In Node A, a matrix is used to record the valid sequence number of the sliding window that can be sent. For each sequence number in the matrix, there are 4 items: packet number, sequence number, “sent or not” and “received ACK or not”. Packets with valid sequence number based on the sliding window will be sent to Interface 1. When receiving ACK from Interface 4, the sliding window and the matrix will be updated and the subsequent packets with new valid sequence number can be sent to Interface 1. In the design, there is a timeout method in Node A and when the timeout is triggered, the packets in the matrix without receiving ACK will be sent again.

In Node B, we also use a matrix to record the valid sequence number of receiving packets, but the matrix is a little different from the one of Node A. For each sequence number of the matrix in Node B, there are only 2 items: sequence number and “received or not”. When receiving a packet from Interface 2, Node B will check if the sequence number of the packet is in the matrix, if it is, the sliding window and the matrix will be updated and the ACK of the received packets will be sent to Interface 3, otherwise it will ignore the packet.

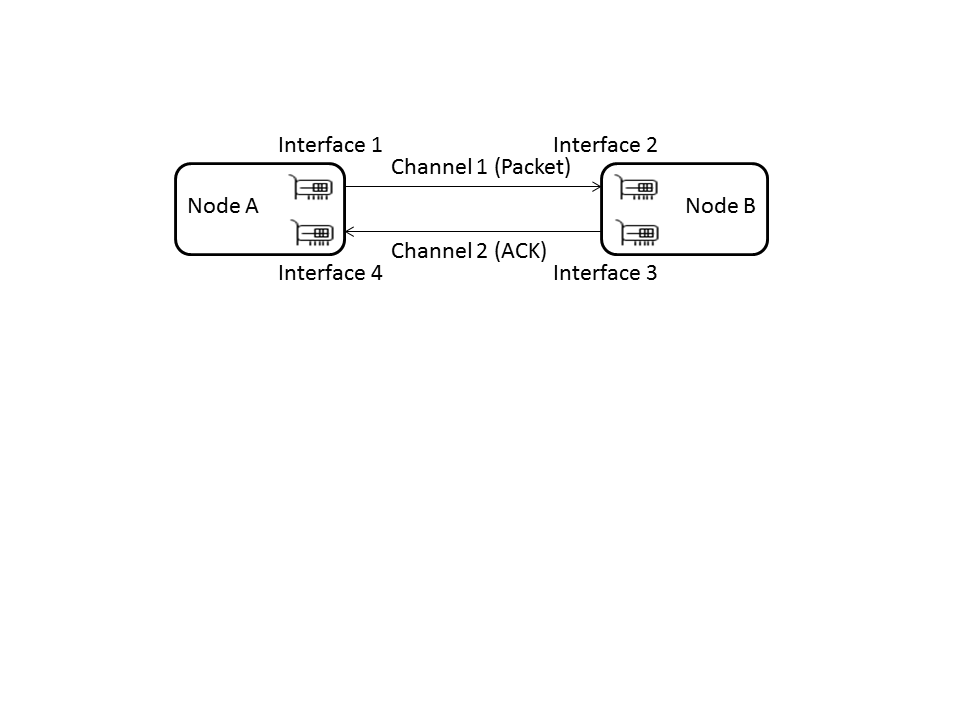


figure 1 Model Structure

1. **Description of Application**

In the program (figure 2), channel and interface are used to synchronize packets and ACKs between Node A and Node B.

The details of the flow of Node A and Node B are shown below:

Flow of Node A:

Call Method TimeOut

Update initial matrix of sequence number

Call the method to send packets based on matrix

while True:

Read ACK response from Interface 4

Update matrix to mark the ACK of corresponding sequence number to 1

Update the matrix to remove the sequence number which has received ACK and add new sequence number which can be sent

Calculate which packet will be dropped and mark “sent or not” of the packet in the matrix to 1

Call the method to send packets

Flow of Node B:

Update initial matrix of sequence number

while True:

Read packets from interface 2

Update matrix to mark the corresponding sequence number whose packet has been received to 1

Calculate the sequence number of ACK which needs to be sent and put the sequence number into array ack\_seq\_no

Update the matrix to remove the sequence number whose packet has been received

Add new sequence number to the matrix

Calculate which ACK packet will be lost and remove the corresponding sequence number from array ack\_seq\_no

if ack\_seq\_no is larger than 0, call the method to send ACK

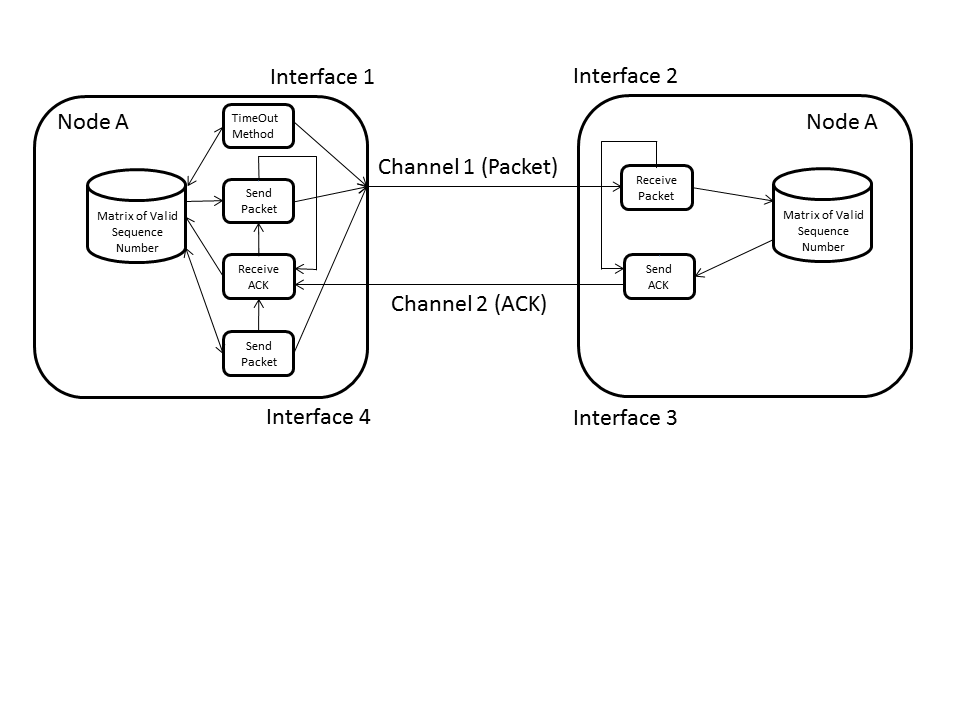


figure 2 Design Structure

1. **Results of Simulation**

The logs of simulation result of some scenarios are attached below (The parameters of these scenarios are listed in table 1):



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parameter | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 |
| Frame Size (bytes) | 10 | 512 | 10 | 10 | 10 |
| Sending Window Size | 10 | 10 | 15 | 10 | 10 |
| Receiving Window Size | 10 | 10 | 5 | 10 | 10 |
| Round Trip Time (ms) | 100 | 100 | 100 | 1000 | 100 |
| Sending Packet Lost Probability (%) | 10 | 10 | 10 | 10 | 20 |
| Sending ACK Lost Probability (%) | 10 | 10 | 10 | 10 | 30 |

table 1 Simulation Scenarios

From the logs, the printout message shows the correct behavioral of sliding window protocol. The followings will explain the meaning of the message of the log files.

1. In the beginning of the log, the parameters of the simulation will be displayed.

Meaning of each parameter:

rtime: Simulation Time based on second

sperror: Sending Packet Lost Probability based on percentage

rperror: Sending ACK Lost Probability based on percentage

fsize: Frame size of Packet and ACK

rtt: Round Trip Time (ms) based on millisecond

rws: Receiving Windows Size

sws: Sending Window Size

######################################################

Run Arguments:

rtime : 10

sperror : 10

rperror : 10

fsize : 10

rtt : 100

rws : 10

sws : 10

######################################################

1. In the beginning of the log, the whole valid sequence number will be shown.

######################################################

[NodeA]Supported seqNo: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

[NodeB]Supported seqNo: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

######################################################

1. The following message in the log shows the parameters of sliding window.

LFR means the Last Acknowledgment Received sequence number, so if the whole valid sequence numbers are from 0 to 19 and the sequence numbers in current sliding window are from 0 to 9, the last acknowledgement received sequence number (LFR) should be 19. The same situation is on LAR.

######################################################

[NodeB]RWS:10 LFR:19 LAF:9

[NodeA]SWS:10 LAR:19 LFS:9

[NodeB]RWS:10 LFR:5 LAF:15

[NodeA]SWS:10 LAR:4 LFS:14

######################################################

1. In the log, when a packet or ACK will be dropped, the message like the following will be displayed.

######################################################

[NodeA][SEND][PKT][DROP][0.000631093978882]: pktNo:6 seqNo:6

[NodeB][SEND][ACK][DROP][0.0534331798553]: seqNo:5

######################################################

1. When sending packets or ACKs, the following message will be shown to display the sequence number, packet number and packet size of the packet of ACK.

######################################################

[NodeB][RECV][PKT][0.0519580841064]: BytesCnt:10 seq\_no:00 pktNo:00000000

[NodeB][RECV][PKT][0.0519580841064]: BytesCnt:10 seq\_no:01 pktNo:00000001

[NodeB][SEND][ACK][0.0545861721039]: BytesCnt:10 seqNo:00

[NodeB][SEND][ACK][0.0545861721039]: BytesCnt:10 seqNo:01

######################################################

1. In Node A, there is a timeout counter and when timeout reaches, Node A will send the packets in the matrix again.

######################################################

[NodeA][SENT][TIMEOUT][0.251137971878]: 1

[NodeA][SENT][TIMEOUT][0.505548000336]: 2

######################################################

1. **Conclusion**

In the design, we realize a sliding window based on a 2 nodes (Node A and Node B) system, in which Node A has a sending sliding window and Node B has a receiving sliding window. Node A will send packets based on the sending sliding window and it will update the sliding window when ACK is received. Node B receives the packets from Node A and it will update the receiving window when a packet is received. After receiving packets, Node B will check if the sequence numbers of packets are continuous following LFR and if they are continuous, Node B will send ACKs of these sequence numbers to Node A.

In the simulation, the sliding windows of Node A and Node B can cooperate correctly and shows the correct result of the behavior of sliding window. Parameters like Round Trip Time, sliding window size, etc. can be parsed through arguments and we can analyze different situations with different parameters which shows a clear behavior of sliding windows.

1. **Run the Program**

The following shows the run command and the arguments we can parse to the command.

Run:

Usage: python TestNetSim.py [options]

Options:

-h, --help show this help message and exit

-t RTT, --rtt=RTT Round Trip Time (ms)

-s SWS, --sws=SWS Sending Window Size

-r RWS, --rws=RWS Receiving Window Size

-f FSIZE, --fsize=FSIZE

Framing Size

-p SPERROR, --sperror=SPERROR

Sending Probability Error

-q RPERROR, --rperror=RPERROR

Receiving Probability Error

-e RTIME, --rtime=RTIME

Run Time: -1:infinite n(n>0):n seconds